Northeastern University - Seattle

Khoury College of Computer Sciences

CS6200 Information Retrieval Fall 2019

Lecture 4:
Transforming Data

Sep 23, 2019

Administrivia

- Quiz 1 evaluated, check it out. Thanks Kartik!
- Quiz 3 will be released today.
- Assignment 1:
  - How's it going? Started on it?
  - Questions?
- Feedback: Quizzes and assignments:
  - Helping you learn?
  - Are they fun?
- Office Hours: do use help from Kartik & me
- Remember:
  - Assignment 1 and Quiz 3 due next Monday: by 9am, Sep 30th

TA Feedback

1. Please describe when asked. It reflects your thought process and help the grader understand what you intend to say.
2. Read the question carefully. A lot of students just discussed one information need when asked for two.
3. For some students it felt like the answers were written just to have something for the submission.
4. Proper formatting the answer will help the grader to understand faster.
5. Avoid late submissions.

Some students have not submitted Quiz 2 responses.

Quiz 2 Discussion .. 1

1. **HTTP 301:** Moved Permanently, for permanent URL redirection. Current links using the URL should be updated. New URL provided in response.
   HTTP 302: Found (Moved Temporarily). [Temporary] URL redirection. Now superseded by 303 and 307.

418 I'm a teapot (RFC 2324, RFC 7168)

This code was defined in 1998 as one of the traditional IETF April Fools' jokes, in RFC 2324, *Hyper Text Coffee Pot Control Protocol*, and is not expected to be implemented by actual HTTP servers. The RFC specifies this code should be returned by teapots requested to brew coffee.[53] This HTTP status is used as an Easter egg in some websites, including Google.com.[54][55]

2. Data normalization: convert all data to a standard form (text, images, video, audio…) while retaining original formatting and making it possible to index and retrieve the text.
3. UTF-8 may be preferred to UTF-32 when storage space and NOT speed of processing is an issue.
4. The two documents have the same checksum value. We could use methods that take into account the positions of bytes, such as cyclic redundancy check (CRC).

Quiz 2 Discussion .. 2

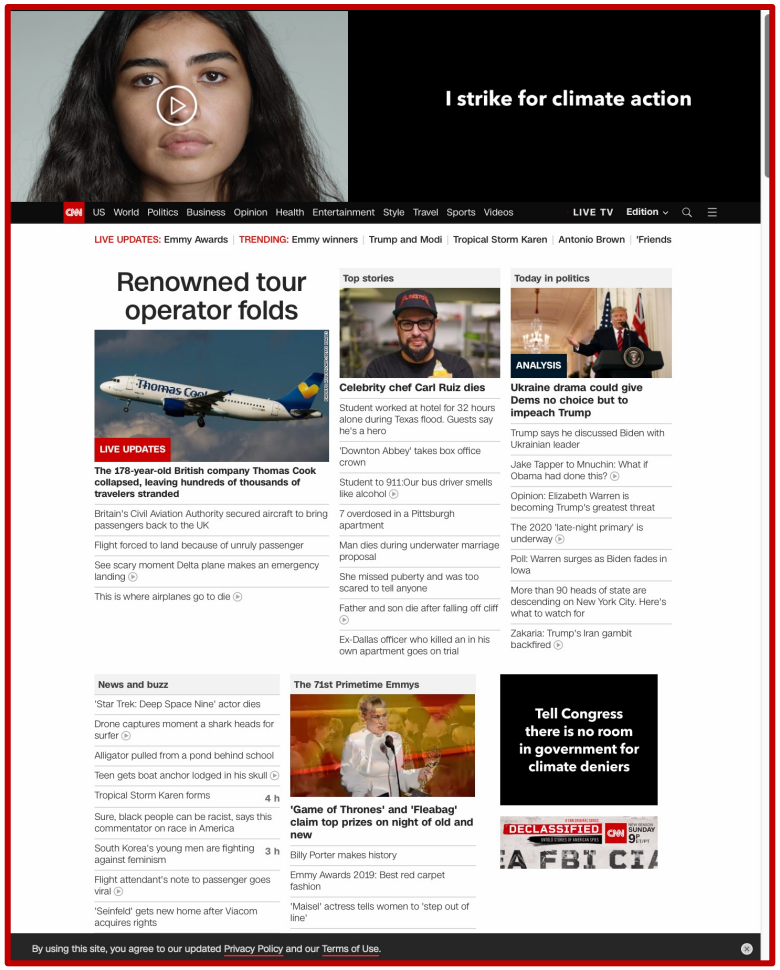
5. Site: CNN.com.

Noise type 1 (text): in the bottom there is a banner bar saying “By using this site, you agree to our updated Privacy Policy and our Terms of Use.”

Approach: find main content block by analyzing the distribution of HTML tags density. This “noise” belongs to a section in the bottom of the HTML page with high density of HTML tags

Noise type 2 (link): Google ads. e.g. https://www.googleadservices.com/pagead/XXX.

Approach: identify a list of ad services host names



Overview

Text Statistics: Zipf's Law & Heaps' Law

Today we make magic !

Dealing with Tokenizing, Stopwords & Stemming

Phrases & N-grams

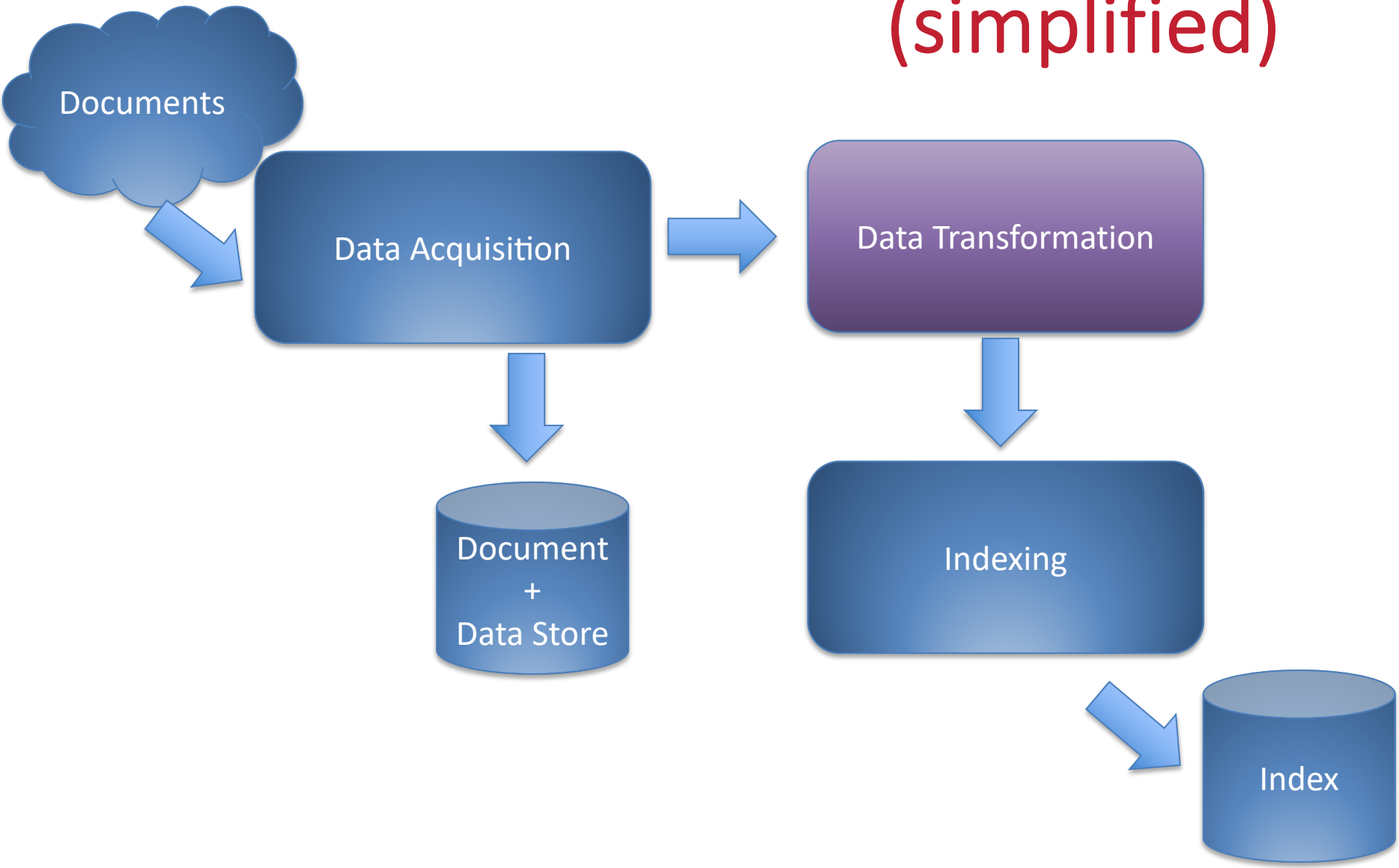
Markup

Link Analysis

Information Extraction

Internationalization (I18n)

Search Engine: Behind the Scenes (simplified)



Processing Data .. 1

Focus on Text now

Prelude to indexing words or 'terms'

- Converting documents to *index terms*

Why 'process' data ?

- Not enough to do an exact string match of user's query
  - not great for effectiveness

Processing Data .. 2

Why 'process' data ? [cont'd]

- Not all words are of equal utility in a search
  - compare the word 'the' with the word 'architecture'
- Sometimes not clear where words begin and end
  - Not even clear what a word is in some languages
    - e.g., Chinese, Korean, Indic languages

TEXT STATISTICS

Text Statistics

Many statistical characteristics of word occurrences are predictable

e.g. distribution of word counts

Retrieval models and ranking algorithms depend heavily on statistical properties of words

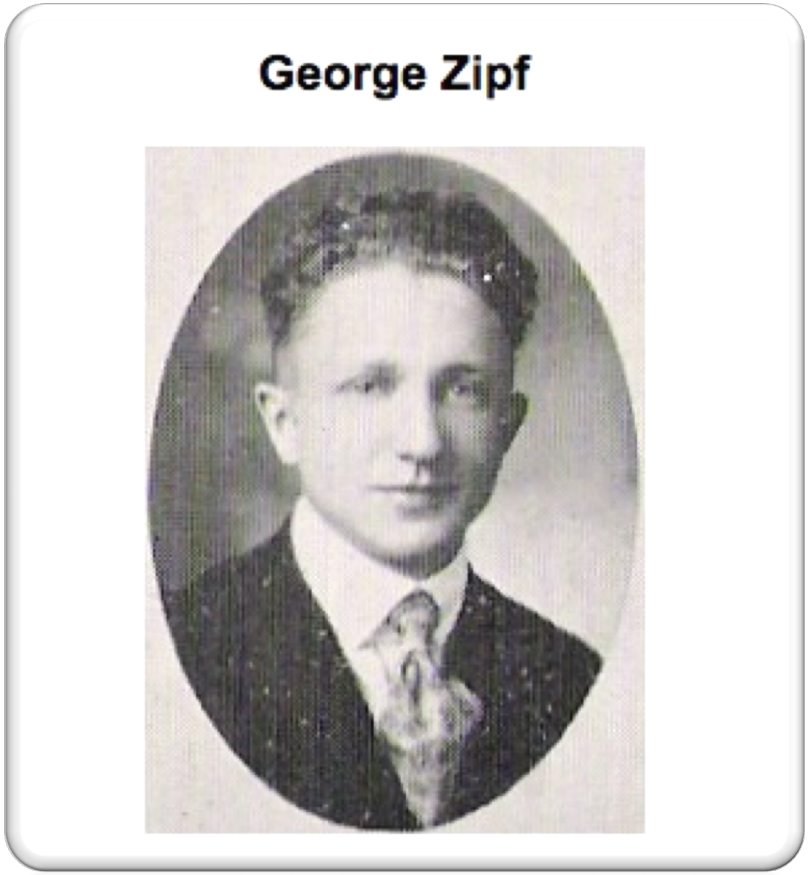
e.g. important words occur often in specific documents ('aboutness') but are not high frequency in collection

We will see more of this in later lectures

ZIPF'S LAW

Zipf's Law .. 1

- Distribution of word frequencies is very *skewed*
  - a few words occur very often, many words hardly ever occur
  - e.g., two common words ("the", "of") make up about 10% of all word occurrences in text documents [~7% and 3.5% of Brown **corpus** of 1m words]



https://en.wikipedia.org/wiki/George_Kingsley_Zipf#/media/File:George_Kingsley_Zipf_1917.jpg

Zipf's Law .. 2

Zipf's "law":
If you take all the words in a collection ('corpus'), count their frequency, and list them in rank order (highest frequency to lowest),

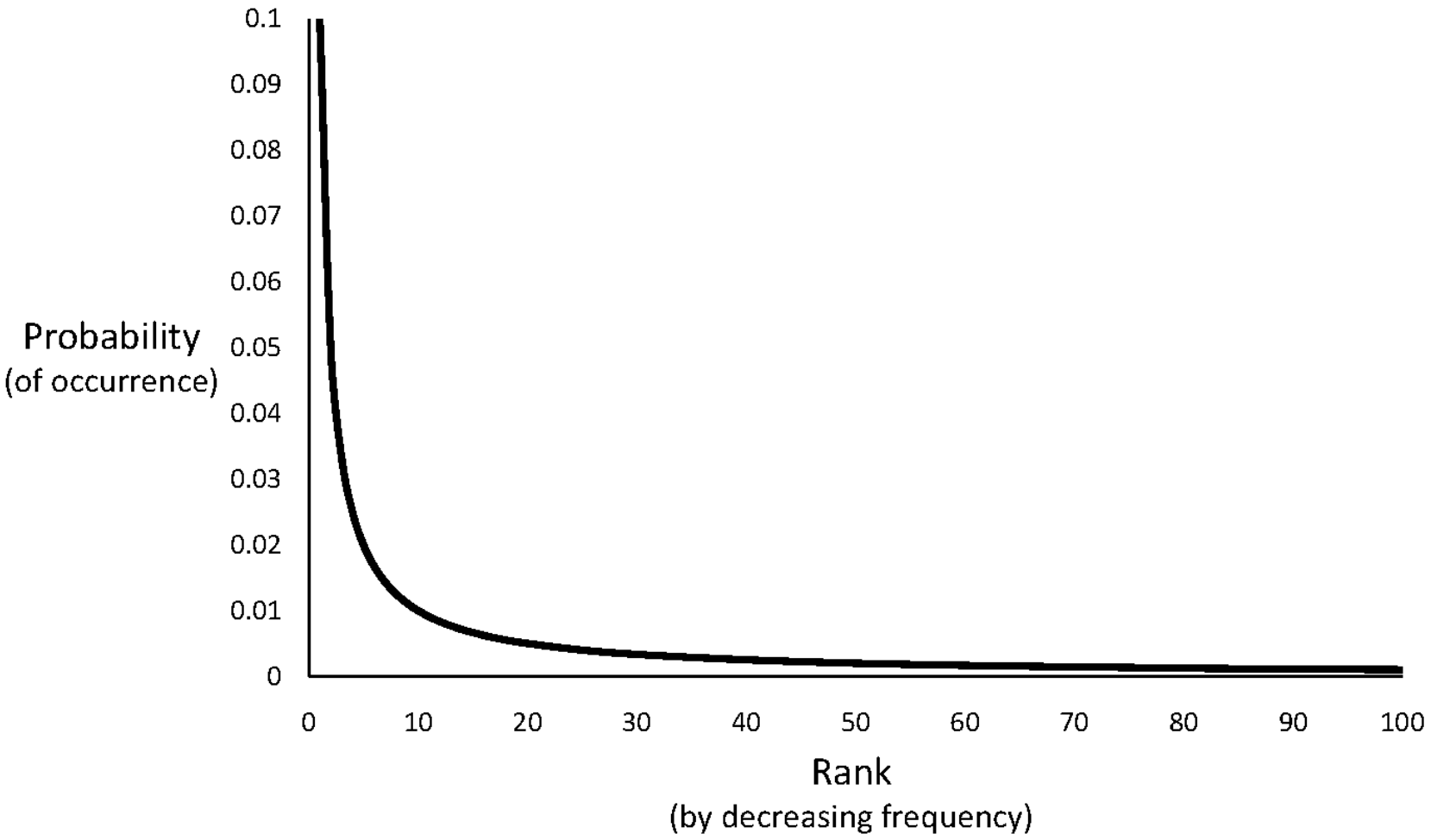
the rank (r) of a word times its frequency (f)
is approximately a constant (k)

– i.e., $r.f \approx k$
or

$$r.P_r \approx c$$

where P_r is probability of word occurrence
and $c \approx 0.1$ for English

Zipf's Law .. 3


0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Probability
(of occurrence)
0
10
20
30
40
50
60
70
80
90
100
Rank
(by decreasing frequency)

News Collection (AP89) Statistics

Hapax Legomena

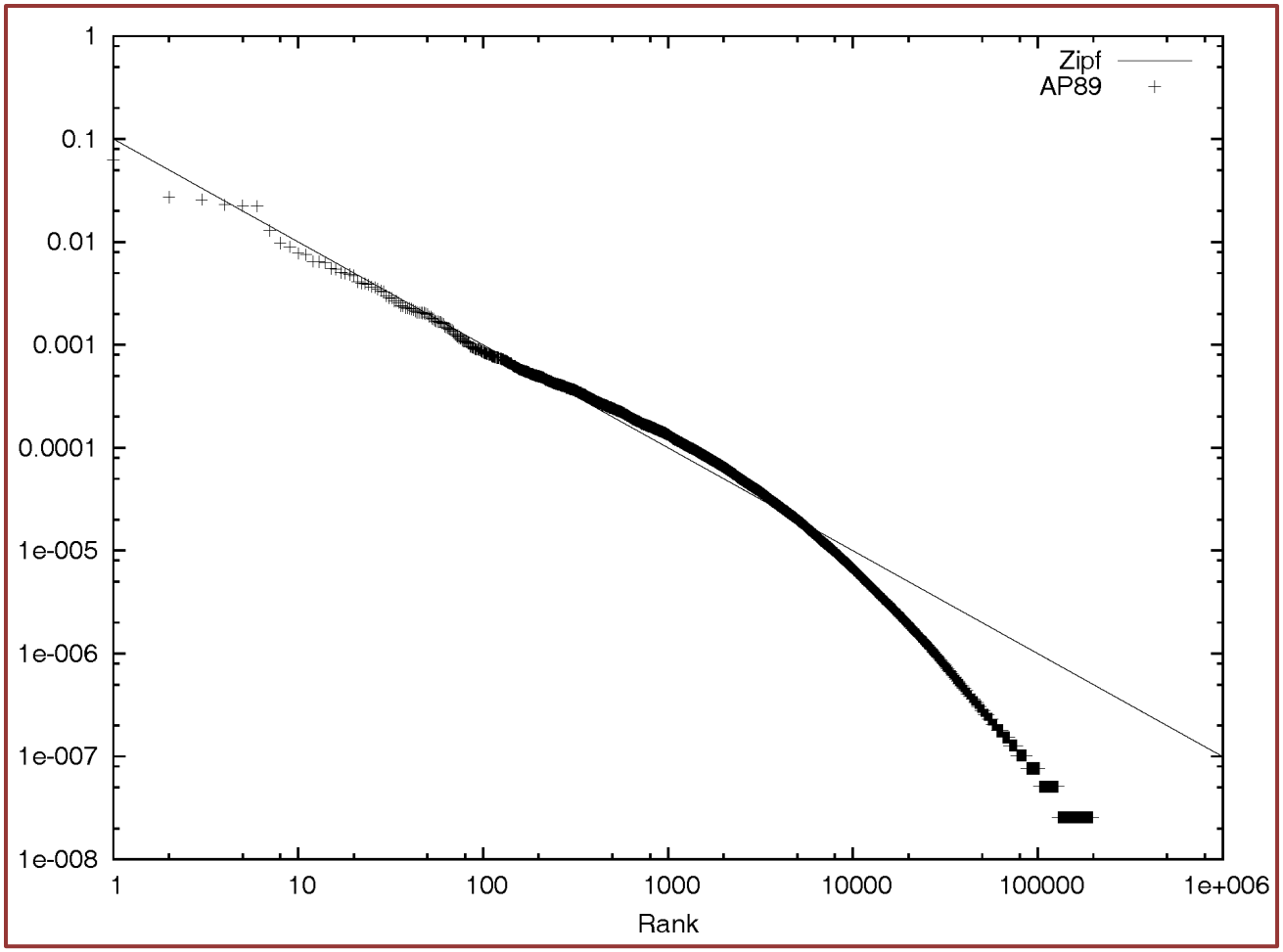
Total documents	84,678
Total word occurrences	39,749,179
Vocabulary size (unique words)	198,763
Words occurring > 1000 times	4,169
Words occurring only once	70,064

Word	Freq.	r	Pr(%)	r.Pr
assistant	5,095	1,021	.013	0.13
sewers	100	17,110	2.56×10^{-4}	0.04
toothbrush	10	51,555	2.56×10^{-5}	0.01
hazmat	1	166,945	2.56×10^{-6}	0.04

Top 50 Words from AP89

Word	*Freq.*	*r*	P_r(%)	$r.P_r$	*Word*	*Freq*	*r*	P_r(%)	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

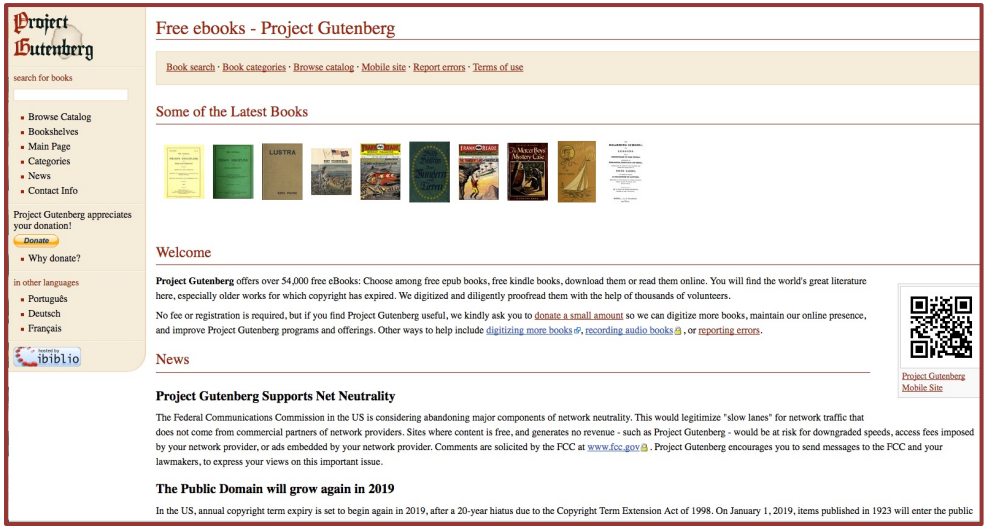
Zipf's Law for AP89



Note problems at high and low frequencies

Fun with Unix & word distributions

First get some text to play with!



Project Gutenberg's *The Adventures of Sherlock Holmes*, by Arthur Conan Doyle, 1892

https://www.gutenberg.org/cache/epub/1661/pg1661.txt

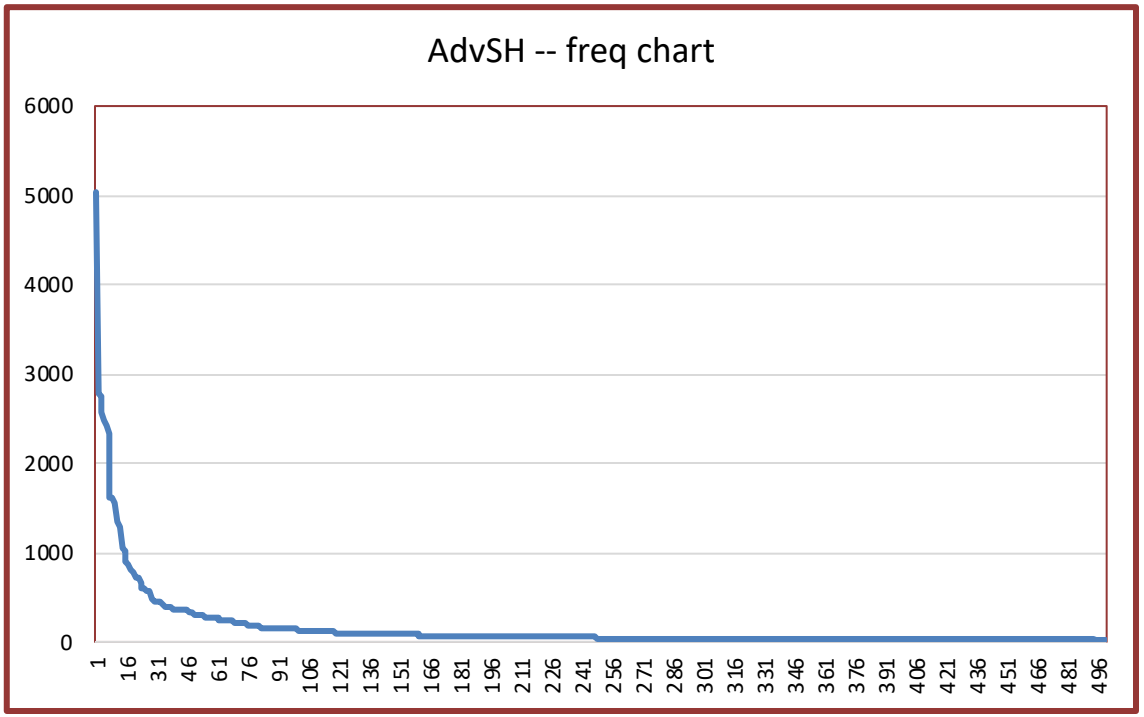
Word Frequencies: quick histogram

```
cat AdventuresSherlockHolmes.txt
   | tr -s ' ' '\n'
    | tr -d "[:punct:]"
     | tr 'A-Z' 'a-z'
      | sort
       | uniq -c
        |sort -rn
         > freq.txt
```

Word Frequencies … example

```
5036 the
2783 i
2749 and
2575
2472 to
2435 of
2345 a
1623 that
1605 in
1544 it
1347 he
1334 you
1282 was
1042 is
```

Histogram: Top 500 terms



Zipf's Law .. 4

What is the proportion of words
with a given frequency?

1. Let last word that occurs *n* times have rank $r_n = k/n$
2. Number of words with frequency *n* is

$$r_n - r_{n+1} = k/n - k/(n+1) = k/n(n+1)$$

3. Proportion of words with frequency n = ($k/n(n+1)$ divided by total number of words)
4. last word with frequency 1 (highest rank = *k)* gives total number of words = k/1 = k
5. and so, proportion with frequency *n* is **$1/n(n+1)$**

Zipf's Law .. 5

- Example word frequency ranking
- To compute number of words with frequency 5,099:

  compute rank of "chemical" minus the rank of "summit"

  (1006 – 1002) = 4

Rank	*Word*	*Frequency*
1000	concern	5,100
1001	spoke	5,100
1002	summit	5,100
1003	bring	5,099
1004	star	5,099
1005	immediate	5,099
1006	chemical	5,099
1007	african	5,098

Predicting the proportion of TREC* word occurrences

- Proportions of words occurring *n* times in 336,310 TREC documents
- Vocabulary size is 508,209

Number of Occurrences (n)	*Predicted Proportion (1/n(n+1))*	*Actual Proportion*	*Actual Number of Words*
1	.500	.402	204,357
2	.167	.132	67,082
3	.083	.069	35,083
4	.050	.046	23,271
5	.033	.032	16,332
6	.024	.024	12,421
7	.018	.019	9,766
8	.014	.016	8,200
9	.011	.014	6,907
10	.009	.012	5,893

HEAPS' LAW

Heaps' Law ..1

Related to Vocabulary Growth

As corpus grows, so does vocabulary size

Fewer new words when corpus is already large

New words come from a variety of sources

names, spelling errors, invented words (e.g. product, company names), code, other languages, email addresses, etc.

examples?

Heaps' Law ..2

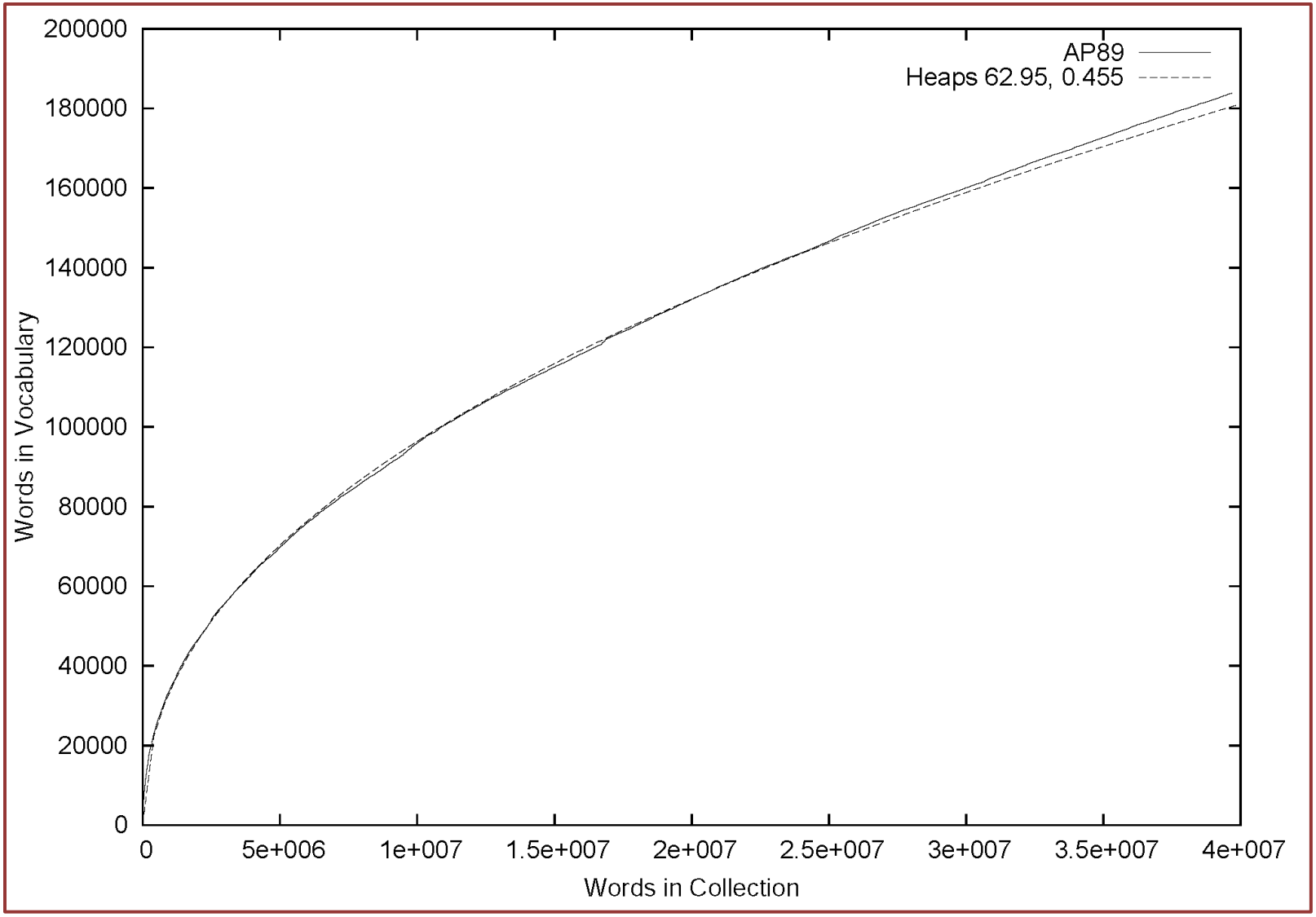
Heaps' Law (aka Herdan's Law):

$$V_R(n) = K * n^{\beta}$$

where

$V_R(n)$ is vocabulary size (number of unique words) in a collection of total n words	*n* is the number of words in corpus,	*K*, β are parameters that vary for each corpus; typical values: $10 \leq K \leq 100$, $0.4 \leq \beta \leq 0.6$

AP89 Example



Heaps' Law Predictions

Predictions for TREC collections :

e.g., When first 10,879,522 words of the AP89 collection scanned

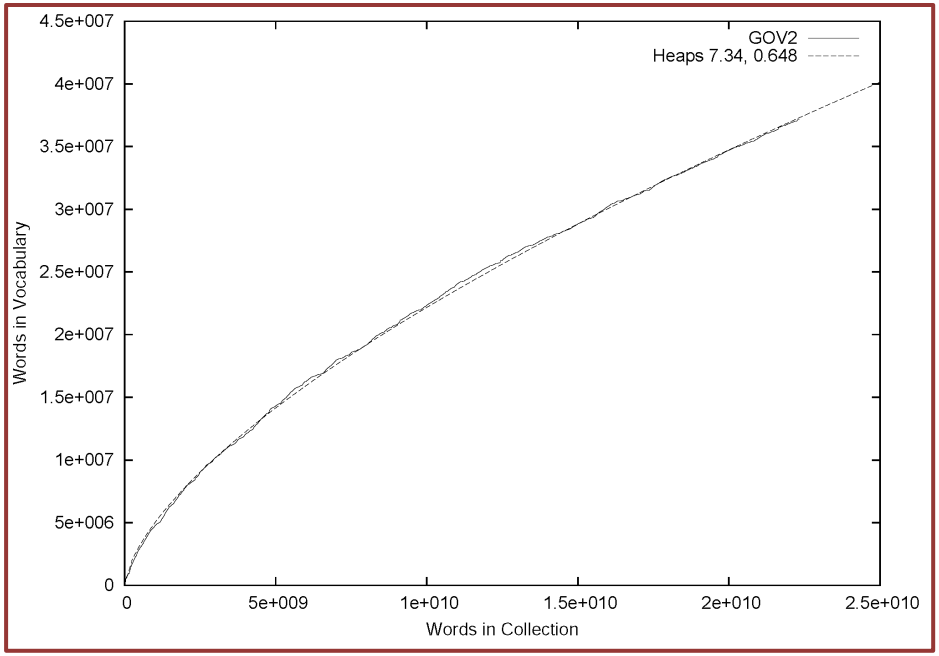
prediction: 100,151 unique words

actual number: 100,024

Predictions for small numbers of words (i.e. < 1000) not as good

GOV2 (Web) Example

- Holds even for very large collections (20 billion words, 2004)
- Search engines must deal with these large and growing vocabularies



Estimating Result Set Size

- How many pages contain *all* of the query terms?
- For the query "*a b c*": P(*a* ∩ *b* ∩ *c*) = *P(a)*P(b)*P(c)*

$$f_{abc} = N \cdot f_a/N \cdot f_b/N \cdot f_c/N = (f_a \cdot f_b \cdot f_c)/N^2$$

Assuming that terms occur independently

- f_{abc} is the estimated size of the result set
- f_a, f_b, f_c are the number of documents that terms *a*, *b*, and *c* occur in (document occurrence frequency)
- *N* is the number of documents in the collection

GOV2 Example

Collection size (*N*) is 25,205,179

Word(s)	*Document Frequency*	*Estimated Frequency*
tropical	120,990	
fish	1,131,855	
aquarium	26,480	
breeding	81,885	
tropical fish	18,472	5,433
tropical aquarium	1,921	127
tropical breeding	5,510	393
fish aquarium	9,722	1,189
fish breeding	36,427	3,677
aquarium breeding	1,848	86
tropical fish aquarium	1,529	6
tropical fish breeding	3,629	18

Result Set Size Estimation

Poor estimates because words are not independent

$P(a \cap b \cap c) = P(a \cap b) \cdot P(c|(a \cap b))$

Better estimates possible if co-occurrence information available

$P(c|(a \cap b))$ ~= larger of P(c|b) and P(c|a)

$f_{tropical \cap fish \cap aquarium} = f_{tropical \cap aquarium} \cdot f_{fish \cap aquarium}/f_{aquarium}$

= 1921 · 9722/26480 = 705

$f_{tropical \cap fish \cap breeding} = f_{tropical \cap breeding} \cdot f_{fish \cap breeeding}/f_{breeding}$

= 5510 · 36427/81885 = 2451

tropical fish aquarium	1,529	6
tropical fish breeding	3,629	18

Result Set Estimation

Even better estimates using initial result set

- Estimate is simply ***C/s****, assuming uniform distribution*
  - where *s* is the proportion of the total documents that have been ranked, and *C* is the number of documents found that contain all the query words

Consider, e.g., "tropical fish aquarium" in GOV2
Actual doc. occ. frequency = 1529

- estimate after processing 3,000 out of the 26,480 documents that contain all three words, *C* = 258
- $f_{tropical \cap fish \cap aquarium}$ = 26480 * 258/(3000) = 2,277

- After processing 20% (~ 5300) of the documents,
- $f_{tropical \cap fish \cap aquarium}$ = 1,778

tropical fish aquarium	1,529	6
tropical fish breeding	3,629	18

Estimating Collection Size

Important issue for Web search engines

Simple technique: use independence model

- Given two words *a* and *b* that are independent
- $f_{ab} / N = f_a / N \cdot f_b / N$
- $N = (f_a \cdot f_b) / f_{ab}$
- e.g., for GOV2
  - $f_{lincoln} = 771{,}326$ $f_{tropical} = 120{,}990$ $f_{lincoln \cap tropical} = 3{,}018$
  - $N = (120990 \cdot 771326)/3018 = 30{,}922{,}045$
  - Actual number is 25,205,179

TOKENIZING, STOPWORDS & STEMMING

TOKENIZING

Tokenizing

Forming words from sequence of characters

Surprisingly complex in English, can be harder in other languages

In search, queries must be tokenized the same way documents are tokenized

otherwise ….?

Tokenizing

Small decisions in tokenizing
can have a major impact
on effectiveness of some queries

- Early IR systems:
  - any sequence of alphanumeric characters of length 3 or more
  - terminated by a space or other special character
  - upper-case changed to lower-case

  Example:

  "AT&T's 2007 bi-annual report showed profits rose 10%."

  →

  "AT & T ' s 2007 bi - annual report showed profits rose 10 % . "

  →

  "2007 annual report showed profits rose"
- Too simple for search applications or even large-scale experiments -- too much information lost

Tokenizing Problems

Short words can be important in some queries

xp, ma, pm, el paso, master p, gm, j lo, World War II, Super Bowl LII

Both hyphenated and non-hyphenated forms of many words are common

Sometimes hyphen not essential

- e-bay, wal-mart, active-x, cd-rom, t-shirts

At other times, hyphens should be considered either as part of the word or a word separator

- winston-salem, mazda rx-7, e-cards, pre-diabetes, t-mobile, spanish-speaking

How do we deal with
New York-New Haven Railroad ?

Tokenizing Problems

Special characters are an important part of tags, URLs, code in documents

Capitalized words *may* have different meaning from lower case words

Bush vs. bush, Apple vs. apple, Blue Jays, Jaguar

Apostrophes can be a part of a word, a part of a possessive, or just a mistake

rosie o'donnell, can't, don't, 80's, 1890's, men's straw hats, master's degree, england's ten largest cities, shriner's

- Could remove apostrophe and 'phrase them' → "rosie o donnell"
- Be consistent

Tokenizing Problems

- Numbers can be important, including decimals

  OM-D E-M1 Mark II, United 93, KUOW 94.9, 288358, 007,98.6°F

- Periods can occur in numbers, abbreviations, URLs, ends of sentences, etc.

  I.B.M., Ph.D., 800.555.1212, northeastern.edu, U.N.C.L.E.

- Commas can occur in non-break situations

  € 22,35 , Pop. 25,000

- Unusual combinations/symbols

  i///u, Ke$ha, H.E.R., .Net, C++, C#, 2Pac, M*A*S*H, 2 Chainz

Tokenizing Process .. 1

First step is to use parser to identify appropriate parts of document to tokenize

- e.g. Maybe ignore <body> … </body>

Defer complex decisions to other components, keep tokenization simple

- word is any sequence of alphanumeric characters, terminated by a space or special character, with everything converted to lower-case
- index everything

E.g.: convert 92.3 → 92 3 but use proximity search to find documents with 92 and 3 adjacent

Tokenizing Process .. 2

Similar to simple tokenizing process used in past

Examples of rules used with TREC

Apostrophes in words ignored

- o'connor → oconnor ; bob's → bobs

Periods in abbreviations ignored

- I.B.M. → ibm ; Ph.D. → ph d

At some point, must handle spelling variants:

Theater vs. Theatre, colour vs. color, Straße vs. Strasse

DEALING WITH STOPWORDS & STEMMING

Stopping .. 1

Function words (determiners, prepositions) have little meaning on their own

- High occurrence frequencies in collection
- Treated as *stopwords* (i.e. removed)
  - reduces index space, improves response time and effectiveness
- But can be important in combinations
  - e.g., “to be or not to be”, “The Who”, “The The”

Stopping .. 2

Stopword list can be created from high-frequency words or based on a standard list

Lists are customized for applications, domains, and even parts of documents

e.g., “click” is a good stopword for anchor text

Best policy is to index all words in documents, and decide which words to use at query time

Stemming .. 1

Many morphological variations of words

inflectional (plurals, tenses) e.g. run, ran

derivational (making verbs nouns etc.) e.g. "nominalization"

In most cases, have same or very similar meanings

Stemmers attempt to reduce morphological variations of words to a common stem

usually involves removing suffixes

Can be done at indexing time or as part of query processing

index just stem, or original + stem

use query expansion

Stemming .. 2

Generally a small but significant effectiveness improvement

- can be crucial for some languages
- e.g., 5-10% improvement for English,
  up to 50% for Arabic

Words with the Arabic root ktb

kitab	*a book*
kitabi	*my book*
al**kitab**	*the book*
kitabuki	*your book* (f)
kitabuka	*your book (m)*
kitabuhu	*his book*
kataba	*to write*
ma**ktaba**	*library, bookstore*
ma**ktab**	*office*

Stemming .. 3

Two basic types of stemmers

Dictionary-based: uses lists of related words

Algorithmic: uses program to determine related words

Algorithmic stemmers

suffix-s: remove 's' endings, assuming plural

- e.g., cats → cat, lakes → lake
- Many *false negatives*:
  supply; supplies → supplie
  [Same stem: supply]
- Some *false positives*: up; ups → up
  [Different stems]

False negative: Same stem, not found
False positive: Different stems, but assumed to be same

Porter Stemmer .. 1

Algorithmic stemmer used in IR experiments since the 70s

Consists of a series of rules designed to handle
the longest possible suffix at each step

Effective in TREC

Produces *stems* not *words*

Makes several errors, difficult to modify

Porter Stemmer .. 2

Example step (1 of 5)

Step 1a:

- Replace ***sses*** by ***ss*** (e.g., stresses → stress).
- Delete ***s*** if the preceding word part contains a vowel not immediately before the ***s*** (e.g., gaps → gap but gas → gas).
- Replace ***ied*** or ***ies*** by ***i*** if preceded by more than one letter, otherwise by ***ie*** (e.g., ties → tie, cries → cri).
- If suffix is ***us*** or ***ss*** do nothing (e.g., stress → stress).

Step 1b:

- Replace ***eed***, ***eedly*** by ***ee*** if it is in the part of the word after the first non-vowel following a vowel (e.g., agreed → agree, feed → feed).
- Delete ***ed***, ***edly***, ***ing***, ***ingly*** if the preceding word part contains a vowel, and then if the word ends in ***at***, ***bl***, or ***iz*** add ***e*** (e.g., fished → fish, pirating → pirate), or if the word ends with a double letter that is not ***ll***, ***ss*** or ***zz***, remove the last letter (e.g., falling→ fall, dripping → drip), or if the word is short, add ***e*** (e.g., hoping → hope).
- Whew!

Porter Stemmer .. 3

- Porter2 stemmer addresses some of these issues
- Similar approach used with stemmers for other languages

False positives	*False negatives*
organization/organ	european/europe
generalization/generic	cylinder/cylindrical
numerical/numerous	matrices/matrix
policy/police	urgency/urgent
university/universe	create/creation
addition/additive	analysis/analyses
negligible/negligent	useful/usefully
execute/executive	noise/noisy
past/paste	decompose/decomposition
ignore/ignorant	sparse/sparsity
special/specialized	resolve/resolution
head/heading	triangle/triangular

Krovetz Stemmer

- Hybrid algorithmic-dictionary
  - Word checked in dictionary
    - If present, either left alone or replaced with “exception”
    - If not present, word is checked for suffixes that could be removed
    - After removal, dictionary is checked again
- Produces words not stems
- Comparable effectiveness
- Lower false positive rate, somewhat higher false negative

Robert (“Bob”) Krovetz
https://www.facebook.com/bob.krovetz

Stemmer Comparison

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Porter stemmer:
document describ market strategi carri compani agricultur chemic report predict market share chemic report market statist agrochem pesticid herbicid fungicid insecticid fertil predict sale market share stimul demand price cut volum sale

Krovetz stemmer:
document describe marketing strategy carry company agriculture chemical report prediction market share chemical report market statistic agrochemic pesticide herbicide fungicide insecticide fertilizer predict sale stimulate demand price cut volume sale

IDENTIFYING AND USING PHRASES & N-GRAMS

Phrases .. 1

Many queries are 2-3-word phrases

Phrases are

- More precise than single words
  - e.g., documents containing “black sea” vs. two words “black” and “sea”
- Less ambiguous
  - e.g., “big apple” vs. “apple”
- Can be difficult for ranking
  - e.g., Given query “fishing supplies”, how do we rank documents with
    - exact phrase many times, exact phrase just once, individual words in same sentence, same paragraph, whole document, variations on words?

Phrases .. 2

How are phrases identified?

Three approaches:

- Identify syntactic phrases using a *part-of-speech* (POS) tagger
- Use word *n-grams*
- Store word positions in indexes and use *proximity operators* in queries

Part of Speech (POS) Tagging

Recall tagging words as nouns, verbs, adjectives etc.?

Phrases can then be defined as simple noun groups, for example

Noun Phrase:
Determiner* {Adjective} * Noun

POS taggers use statistical models of text to predict syntactic tags of words

Example tags:

NN (singular noun), NNS (plural noun), VB (verb), VBD (verb, past tense), VBN (verb, past participle), IN (preposition), JJ (adjective), CC (conjunction, e.g., “and”, “or”), PRP (pronoun), and MD (modal auxiliary, e.g., “can”, “will”).

POS Tagging Example: Brill Tagger

Original text:

Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:

Document/NN will/MD describe/VB marketing/NN strategies/NNS carried/VBD out/IN by/IN U.S./NNP companies/NNS for/IN their/PRP agricultural/JJ chemicals/NNS ,/, report/NN predictions/NNS for/IN market/NN share/NN of/IN such/JJ chemicals/NNS ,/, or/CC report/NN market/NN statistics/NNS for/IN agrochemicals/NNS ,/, pesticide/NN ,/, herbicide/NN ,/, fungicide/NN ,/, insecticide/NN ,/, fertilizer/NN ,/, predicted/VBN sales/NNS ,/, market/NN share/NN ,/, stimulate/VB demand/NN ,/, price/NN cut/NN ,/, volume/NN of/IN sales/NNS ./.

Eric Brill
From https://news.ucsc.edu/2012/10/research-review-day.html

Sample Noun Phrases (TREC, Patents)

TREC data		Patent data	
Frequency	*Phrase*	*Frequency*	*Phrase*
65824	united states	975362	present invention
61327	article type	191625	u.s. pat
33864	los angeles	147352	preferred embodiment
18062	hong kong	95097	carbon atoms
17788	north korea	87903	group consisting
17308	new york	81809	room temperature
15513	san diego	78458	seq id
15009	orange county	75850	brief description
12869	prime minister	66407	prior art
12799	first time	59828	perspective view
12067	soviet union	58724	first embodiment
10811	russian federation	56715	reaction mixture
9912	united nations	54619	detailed description
8127	southern california	54117	ethyl acetate
7640	south korea	52195	example 1
7620	end recording	52003	block diagram
7524	european union	46299	second embodiment
7436	south africa	41694	accompanying drawings
7362	san francisco	40554	output signal
7086	news conference	37911	first end
6792	city council	35827	second end
6348	middle east	34881	appended claims
6157	peace process	33947	distal end
5955	human rights	32338	cross-sectional view
5837	white house	30193	outer surface

Word N-Grams

POS tagging too slow for large collections

Simpler definition:
phrase is any sequence of
n words – known as *n-grams*

unigram: single words,
bigram: 2 word sequence,
trigram: 3 word sequence

N-grams also used at character level for applications such as OCR

N-grams typically formed from *overlapping* sequences of words

e.g. move n-word "window" one word at a time in document

N-Grams

Frequent n-grams more likely to be meaningful phrases

N-grams form a Zipf-ian distribution

Better fit than words alone

Could index all n-grams up to specified length

Much faster than POS tagging

Uses a lot of storage

- e.g., document of 1,000 words has 3,990 word n-grams of length $2 \leq n \leq 5$

Google N-Grams

- Web search engines index n-grams, with freq >= 40
- Most frequent trigram in English is "all rights reserved"
- In Chinese, "limited liability corporation"

https://research.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html

Google sample

Number of tokens:	1,024,908,267,229
Number of sentences:	95,119,665,584
Number of unigrams:	13,588,391
Number of bigrams:	314,843,401
Number of trigrams:	977,069,902
Number of fourgrams:	1,313,818,354
Number of fivegrams:	1,176,470,663

DEALING WITH MARKUP

Document Structure and Markup

Some parts of documents are more important than others

Document parser recognizes structure using markup, such as HTML tags

Headers, anchor text, bolded text all likely to be important

- can be weighted differently <title/> more important than <h1/>, <b/> important etc.

Metadata can also be important

Links used for *link analysis*

Example Web Page .. 1

Tropical fish

From Wikipedia, the free encyclopedia

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species. Fishkeepers often use the term *tropical fish* to refer only those requiring fresh water, with saltwater tropical fish referred to as *marine fish*.

Tropical fish are popular aquarium fish , due to their often bright coloration. In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Example Web Page .. 2

```
<html>
<head>
<meta name="keywords" content="Tropical fish, Airstone, Albinism, Algae eater,
Aquarium, Aquarium fish feeder, Aquarium furniture, Aquascaping, Bath treatment
(fishkeeping),Berlin Method, Biotope" />
…
<title>Tropical fish - Wikipedia, the free encyclopedia</title>
</head>
<body>
 …
<h1 class="firstHeading">Tropical fish</h1>
…
<p><b>Tropical fish</b> include <a href="/wiki/Fish" title="Fish">fish</a> found in <a
href="/wiki/Tropics" title="Tropics">tropical</a> environments around the world,
including both <a href="/wiki/Fresh_water" title="Fresh water">freshwater</a> and <a
href="/wiki/Sea_water" title="Sea water">salt water</a> species. <a
href="/wiki/Fishkeeping" title="Fishkeeping">Fishkeepers</a> often use the term
<i>tropical fish</i> to refer only those requiring fresh water, with saltwater tropical fish
referred to as <i><a href="/wiki/List_of_marine_aquarium_fish_species" title="List of
marine aquarium fish species">marine fish</a></i>.</p>
<p>Tropical fish are popular <a href="/wiki/Aquarium" title="Aquarium">aquarium</a>
fish , due to their often bright coloration. In freshwater fish, this coloration typically
derives from <a href="/wiki/Iridescence" title="Iridescence">iridescence</a>, while salt
water fish are generally <a href="/wiki/Pigment" title="Pigment">pigmented</a>.</p>
…
</body></html>
```

LINK ANALYSIS

Link Analysis

Links: key component of the Web

Important for navigation, but also for search

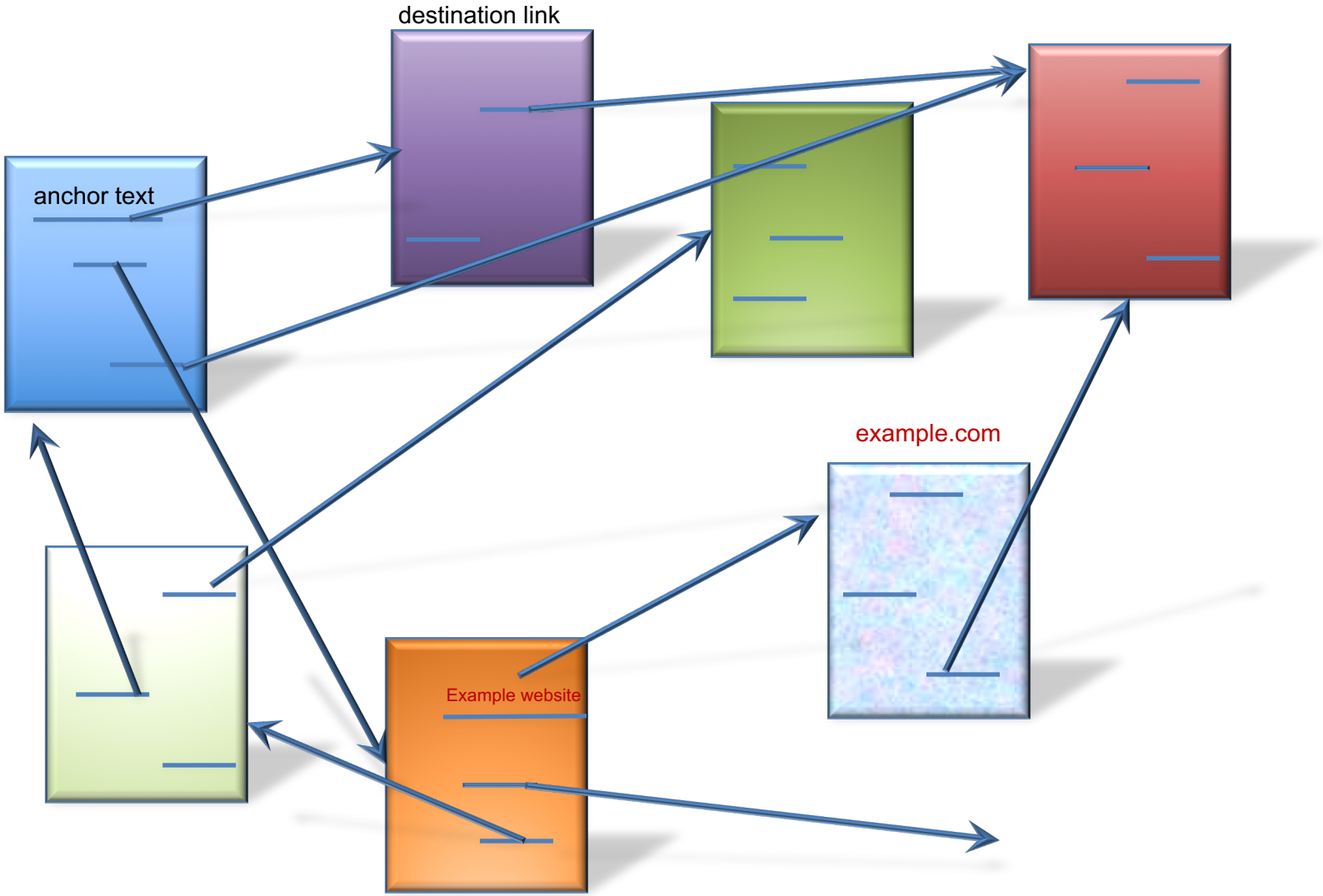
e.g.,
Example website

“Example website” is the anchor text

“http://example.com” is the destination link

Both are used by search engines

Anchor Text & Destination Links



Anchor Text

Anchor text tends to be short, descriptive, very much like query text

Used as a description of the content of the *destination page*

i.e., collection of anchor text in all links pointing to a page used as an additional text field

Anchor text has significant impact on effectiveness for *some types of queries*

PageRank

Billions of web pages, some more informative than others

Links can be viewed as information about the *popularity* (*authority*) of a web page

can be used by ranking algorithm

Inlink count could be used as simple measure

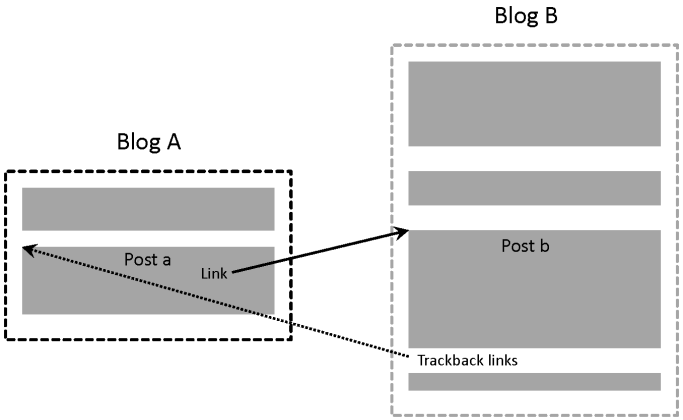
Link analysis algorithms like PageRank provide more reliable ratings

less susceptible to link spam

More about PageRank soon!

Link Quality

- Link quality is affected by spam and other factors
  - e.g., *link farms* to increase PageRank
  - *trackback links* in blogs can create loops



  - links from comments section of popular blogs

INFORMATION EXTRACTION

Information Extraction

Automatically extract structure from text

annotate document using tags to identify extracted structure

Named entity recognition

identify words that refer to something of interest in text e.g., people, companies, locations, dates, product names, prices, etc.

Named Entity Recognition .. 1

- Example shows semantic annotation using XML tags
- Information extraction could include document structure and more complex features such as *relationships* and *events*

Fred Smith, who lives at 10 Water Street, Springfield, MA, is a long-time collector of **tropical fish.**

<PersonName><GivenName>Fred</GivenName> <Sn>Smith</Sn> </PersonName>, who lives at <address><Street >10 Water Street</Street>, <City>Springfield</City>, <State>MA</State></address>, is a long-time collector of tropical fish.

Named Entity Recognition .. 2

- *Rule-based*
  - Uses *lexicons* (lists of words and phrases) that categorize names
    - e.g., locations, peoples' names, organizations, time, currency, etc.
  - Rules also used to verify or find new entity names
    - e.g., "<number> <word> street" for addresses
    - "<street address>, <city>" or "in <city>" to verify city names
    - "<street address>, <city>, <state>" to find new cities
    - "<title> <name>" to find new names
- Rules either developed manually by trial and error or using machine learning techniques

Named Entity Recognition .. 3

- *Statistical*
  - uses a probabilistic model of the words in and around an entity
  - probabilities estimated using *training data* (manually annotated text)
  - One such approach: Hidden Markov Model (HMM)
  - [More recently, Deep Learning approaches used for NER]

HMM for Extraction .. 1

Resolve ambiguity in a word using *context*

e.g., “marathon” is a location or a sporting event, “boston marathon” is a specific sporting event

Model context using a *generative* model of the sequence of words

Markov property: the next word in a sequence depends only on a small number of the previous words

HMM for Extraction .. 2

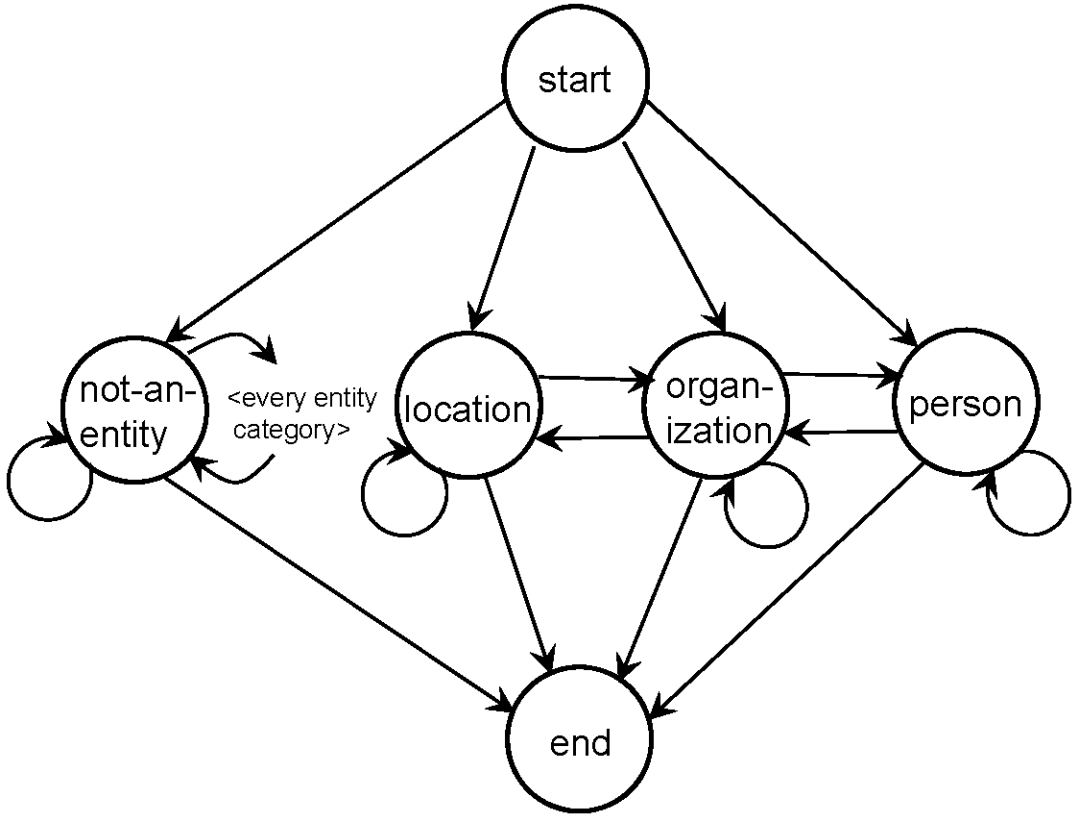
Markov Model describes a process as a collection of states with transitions between them

- each transition has a probability associated with it
- next state depends only on current state and transition probabilities

Hidden Markov Model

- each state has a set of possible outputs
- outputs have probabilities

HMM Sentence Model



- Each state is associated with a probability distribution over words (the output)

HMM for Extraction .. 3

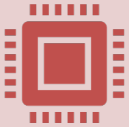
Can generate sentences with this model

To recognize named entities, find sequence of "labels" that give highest probability for the sentence

only the outputs (words) are visible or observed

states are "hidden"

e.g., <start><name><not-an-entity><location><not-an-entity><end>



Viterbi algorithm used for recognition

Named Entity Recognition .. 4

Accurate recognition requires about 1M words of training data (1,500 news stories)

may be more expensive than developing rules for some applications

Both rule-based and statistical can achieve about 90% effectiveness for categories such as names, locations, organizations

others, such as product names, not as easy to identify

INTERNATIONALIZATION (AKA I18N)

Internationalization .. 1

Over 65% of the Web is in English, but …

About 50% of Web users do not use English as their primary language

Most search applications deal with multiple languages

monolingual search: search designed for 1 specific language

cross-language search: search in multiple languages at the same time

Internationalization .. 2

Many aspects of search engines are language-neutral

Major differences:

- Text encoding (converting to Unicode)
- Character normalization
- Tokenizing (what if no word separators)
- Compound words and De-compounding

Character normalization

- Diacritic removal, other special characters
  E.g., **é** = e, **ø** = o (all languages) ...
- Some characters map to multiple alternative characters
  E.g., **ö** = oe or o, **ü** = ue or u (German) ...
- Chinese Simplified (China, Singapore) :e.g. 学
  vs. Traditional characters (Taiwan): 學
- Chinese characters (Kanji) used in Japanese

Issues in I18n: Tokenization

Chinese, Japanese and Thai do not use white spaces to indicate word boundaries

e.g. 莎拉波娃现在居住在美国东南部的佛罗里达
(Maria Sharapova now lives in Florida in southeastern United States)

Two approaches to tokenization for these language

- Character-based tokenization
- Word segmentation
- Combination
- N-gram approaches

A one ‘word’ description of an area near Kanchipuram, Tamil Nadu, India

The world’s longest ‘word’ (195 Sanskrit ‘characters’):

निरन्तरान्धकारितादिगन्तरकन्दलदमन्दसुधारसबिन्दुसान्द्रतरघनाघनवृन्दसन्देहकरस्यन्दमानमकरन्दबिन्दुबन्धुरतरमाकन्दतरुकुलतल्पकल्पमृदुलसिकताजालजटिलमूलतलमरुवकमिलदलघुलघुलयकलितरमणीयपानीयशालिकाबालिकाकरारविन्दगलन्तिकागलदेलालवङ्गपाटलघनसारकस्तूरिकातिसौरभमेदुरलघुतरमधुरशीतलतरसलिलधारानिराकरिष्णुतदीयविमलविलोचनमयूखरेखापसारितपिपासायासपथिकलोकान्

Roman transliteration of this word (428 letters):

nirantarāndhakāritādigantarakandaladamandasudhārasabindusāndrataraghanāghanavr̥ndasandehakarasyandamānamakarandabindubandhurataramākandatarukulatalpakalpamr̥dulasikatājālajaṭilamūlatalamaruvakamiladalaghulaghulayakalitaramaṇīyapānīyaśālikābālikākarāravindagalantikāgaladelālavaṅgapāṭalaghanasārakastūrikātisaurabhameduralaghutaramadhuraśītalatarasaliladhārānirākariṣṇutadīyavimalavilocanamayūkharekhāpasāritapipāsāyāsapathikalokān

Rough translation:

In it, the distress, caused by thirst, to travelers, was alleviated by clusters of rays of the bright eyes of the girls; the rays that were shaming the currents of light, sweet and cold water charged with the strong fragrance of cardamom, clove, saffron, camphor and musk and flowing out of the pitchers (held in) the lotus-like hands of maidens (seated in) the beautiful water-sheds, made of the thick roots of vetiver mixed with marjoram, (and built near) the foot, covered with heaps of couch-like soft sand, of the clusters of newly sprouting mango trees, which constantly darkened the intermediate space of the quarters, and which looked all the more charming on account of the trickling drops of the floral juice, which thus caused the delusion of a row of thick rainy clouds, densely filled with abundant nectar

http://www.hitxp.com/articles/literature/world-longest-word-language-guinness-record/
http://www.guinnessworldrecords.com/world-records/longest-word

Issues in I18n: Compound Words

Compound words in Germanic languages, Korean

Example:
Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz (German) [~=
Beef Labeling Monitoring Delegation Act]

De-compounding

identifies components of compound words and creates tokens for them

allows users to find compound words using components

- Example: q=(Kapitän Patent) matches Kapitänspatent (German – captain's license)

Summary

We looked at some text statistics and applications.

We saw how search engines deal with tokenization, stopwords and stemming, phrases & n-grams.

We peeked at link analysis & information extraction.

We finished with a look at internationalization issues.

Next week: **Indexing and Ranking**

Readings

Croft, Metzler & Strohman (CMS), Chapter 4

Check out TREC (Text Retrieval Conference), especially Overview and Tracks
http://trec.nist.gov/ and the Wikipedia page at
https://en.wikipedia.org/wiki/Text_Retrieval_Conference

Optional

Manning, Raghavan & Schütze (MRS), Chapter 2
You can skip Sections 2.3, 2.4.2, 2.4.3

I like the notion of *Hapax Legomena*. Check out:
https://www.wired.com/2012/01/hapax-legomena-and-zipfs-law/

Check out Google Books N-gram Viewer
https://books.google.com/ngrams

Questions?